CRM 88-129 / October 1988

A Comparison of Models for Determining Spares Requirements for Aircraft Battle Damage Repair

John D. Parsons

CNA A Division of Hudson Institute
CENTER FOR NAVAL ANALYSES
4401 Ford Avenue • Post Office Box 16268 • Alexandria, Virginia 22302-0268

REPORT DOCUMENTATION PAGE

Form Approved
OPM No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources gathering and maintaining the data needed, and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Information and Regulatory Affairs, Office of Management and Budget, Washington, DC 20503.

1. AGENCY USE ONLY *(Leave Blank)*

2. REPORT DATE
October 1988

3. REPORT TYPE AND DATES COVERED
Final

4. TITLE AND SUBTITLE
A Comparison of Models for Determining Spares Requirements for Aircraft Battle Damage Repair

5. FUNDING NUMBERS
C - N00014-87-C-0001
PE - 65154N
PR - R0148

6. AUTHOR(S)
John D. Parsons

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Center for Naval Analyses
4401 Ford Avenue
Alexandria, Virginia 22302-0268

8. PERFORMING ORGANIZATION REPORT NUMBER
CRM 88-~~128~~
129

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Cleared for Public Release; Distribution Unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

As part of the Aircraft Battle Damage Repair (ABDR) project, CNA analysts performed research into finding an appropriate model for determining ABDR spares requirements. The analysis focuses on the relative performance of four spares requirements models, given uncertainties associated with predicting battle damage rates.

14. SUBJECT TERMS
Aircraft maintenance, Battle damage, Computer programs, Damage assessment, Failure, Inventory control, Mathematical analysis, Mathematical models, Numerical methods and processes, Predictions, Probability, Repair, Spare parts

15. NUMBER OF PAGES
63

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT
CPR

18. SECURITY CLASSIFICATION OF THIS PAGE
CPR

19. SECURITY CLASSIFICATION OF ABSTRACT
CPR

20. LIMITATION OF ABSTRACT
SAR

CNA CENTER FOR NAVAL ANALYSES

A Division of Hudson Institute 4401 Ford Avenue • Post Office Box 16268 • Alexandria, Virginia 22302-0268 • (703) 824-2000

17 November 1988

MEMORANDUM FOR DISTRIBUTION LIST

Subj: Center for Naval Analyses Research Memorandum 88-129

Encl: (1) CNA Research Memorandum 88-129, *A Comparison of Models for Determining Spares Requirements for Aircraft Battle Damage Repair*, by John D. Parsons, Oct 1988

1. Enclosure (1) is forwarded as a matter of possible interest.

2. As part of the Aircraft Battle Damage Repair (ABDR) Project, CNA analysts performed research into finding an appropriate model for determining ABDR spares requirements. The analysis focuses on the relative performance of four spares requirements models, given uncertainties associated with predicting battle damage rates.

Peter Evanovich

Peter J. Evanovich
Director
Readiness and Sustainability Program

Distribution List:
Major R. McCammond - OP-514M

A-1

CRM 88-129 / October 1988

A Comparison of Models for Determining Spares Requirements for Aircraft Battle Damage Repair

John D. Parsons

Naval Warfare Operations Division

CNA A Division of Hudson Institute

CENTER FOR NAVAL ANALYSES

4401 Ford Avenue • Post Office Box 16268 • Alexandria, Virginia 22302-0268

ABSTRACT

As part of the Aircraft Battle Damage Repair (ABDR) Project, CNA analysts performed research into finding an appropriate model for determining ABDR spares requirements. The analysis focuses on the relative performance of four spares requirements models, given uncertainties associated with predicting battle damage rates.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

SUMMARY AND RESULTS OF ANALYSIS

INTRODUCTION

As part of the Aircraft Battle Damage Repair (ABDR) Study, CNA analysts performed research into finding an appropriate model for determining ABDR material requirements. This research memorandum documents the analysis and recommends a specific model for determining ABDR spares requirements.

This section of the paper contains a nontechnical discussion of the analysis, a summary of the results, and the conclusion of the analysis. The next section contains technical details, including a discussion of the mathematics behind the analysis and precise definitions of the spares models and probability distributions used in the analysis. Detailed results of the analysis are presented in appendix A. Two of the spares models considered in this analysis had a surprising relationship; Appendix B contains a detailed example that helps to illustrate the reasons for this. A listing of the major computer program used in the analysis is contained in appendix C.

BACKGROUND

The overall modeling process for ABDR material requirements may be broken into two parts: (1) the development data bases for predicting battle damage rates, and (2) development of a spares model to develop ABDR spares lists from these data bases. Current Navy planning calls for these data bases for predicting battle damage rates to be developed by the Naval Weapons Center (NWC) and the Naval Air Development Center (NADC). Essentially, NWC vulnerability and susceptibility models will be used to model aircraft damage; the ABDR material required to repair each type of damage incident is determined in conjunction with NADC data bases. The end result of the process will probably be a collection of component or part-level battle damage rates. Following the development of a data base for predicting battle damage rates, NADC will use a spares model to develop ABDR allowance lists.

CNA's research effort focused only on the spares model and did not consider the methods used to develop the data base for predicting battle damage rates.

The objective of the research effort was to find a spares model appropriate for determining ABDR material requirements.

There are at least two striking differences between the types of failures experienced normally in peacetime and those associated with battle damage: uncertainty and dependence.

The peacetime failure rates from which Aviation Consolidated Allowance Lists (AVCALs) are developed are based on actual usage data. In determining ABDR requirements, planners must use predicted battle damage rates rather than peacetime usage data; of course, there are not any relevant, actual battle damage data for use in this process. The Navy's plans to use vulnerability and susceptibility models to develop predicted battle damage rates seems reasonable; indeed, an alternative method for developing predicted battle damage rates is not evident. However, regardless of how detailed and accurate the models are from an engineering standpoint, the damage-rate predictions will be based in large part on the following information:

- Educated guesses for model parameters, particularly descriptions of threats, tactics, and operational objectives
- Intelligence information
- Historical data from conflicts in which weapons, tactics, and threats were different from those likely to be experienced in a future conflict.

Hence, there is more uncertainty associated with battle damage rates than with empirical, peacetime usage data used to develop AVCAL requirements. This uncertainty should be considered when selecting a model for ABDR material requirements.

Peacetime spares models assume independence of failures. That is, given two distinct items on the aircraft, the probability that item A fails during a flight is independent of the probability that item B fails during the flight. In general, the assumption of the independence of battle damage "failures" is not easy to accept. Indeed, it is clear that the physical proximity of two items is important, and if two items are located close together, they will tend to be damaged together.

Consider the following example of the effect of the independence assumption in the battle damage situation. Assume two parts, A and B, are both required to have a flyable aircraft. Also assume that parts A and B have identical predicted battle damage rates, and part A costs ten times as much as part B. A readiness-based spares model, analogous to the readiness-based AVCAL model currently planned for use in developing F-18 AVCALs, trades off cost against improvements in readiness when computing allowance levels. This type of spares model also assumes the independence of failures. The readiness-based model would tend to stock part B to a higher level than part A. Just for the sake of this example, suppose ten items of part B are stocked and four items of part A are stocked. Assume that parts A and B are located next to each other in the aircraft, so that part A and part B are always damaged together. In this case, six extra items of part B are useless, because any aircraft needing the replacement of part B also requires the replacement of part A.

Clearly, the assumption of independence in a peacetime spares model must be carefully considered if the model is adapted for ABDR spares.

Any model for developing spare parts requirements must use predicted battle damage rates. As discussed above, actual battle damage rates may be different from the predicted rates; moreover, the actual rates may not satisfy assumptions of independence built into certain spares models. Most spares models will do well in determining requirements when the predicted and actual damage rates are equal, but what happens if the predicted rates and actual damages rates are different? The following example, depicted in figure 1, illustrates the idea that ABDR sparing models should have the flexibility to provide support under the uncertainty assumption about the battle damage rate planning factors. Suppose two different models are used to form ABDR spare parts inventories, A and B respectively, from the same set of predicted damage rates. Suppose stockpile A performs much better than stockpile B when the actual damage rates and the predicted damage rates are equal, but as actual damage rates and predicted damage rates become further apart, stockpile B's performance becomes better than stockpile A's. Stockpile A, optimized for the situation in which predicted and actual damage rates are equal, is not flexible enough to cope well with different damage rates. Stockpile A performs well only in a specific, narrow setting. Stockpile B, designed for flexibility, performs reasonably well in a wide range of situations but is not the optimal model when predicted and actual damage rates are equal.

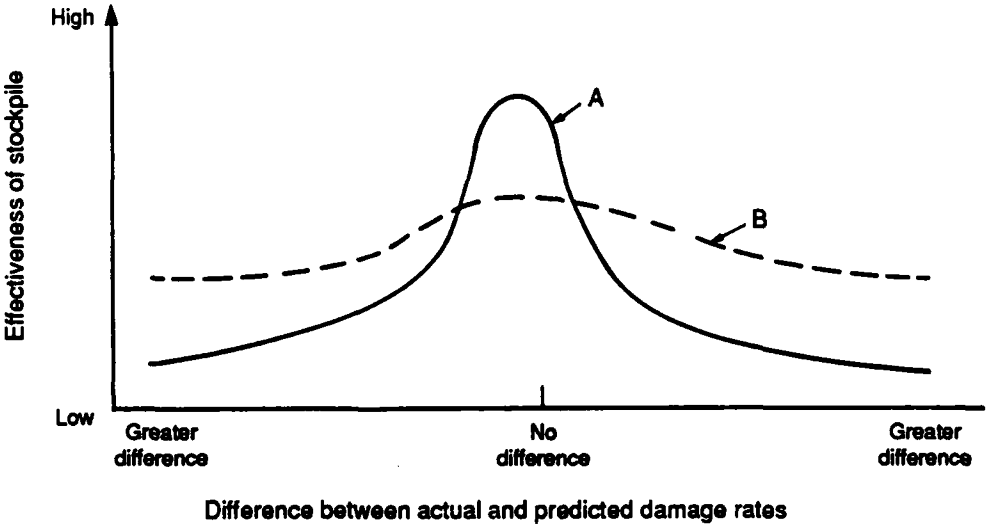

FIG. 1: FLEXIBILITY VERSUS OPTIMALITY

NONTECHNICAL DEFINITIONS OF SPARES MODELS

Four alternative spares models were considered in this analysis: the DBS-Part, RBS-Part, DBS-Damage Incident, and RBS-Damage Incident models.[1] Given a list of candidate parts for inclusion in a spare parts stockpile, each spares model is designed to develop a stock level for each candidate part sufficient to cover demands over a user-specified period of time under the assumption of no resupply or repair of the damaged part.

The list of candidate parts does not necessarily include all parts on the aircraft. For example, the Navy may decide to to include only items that are not normally required to support peacetime flight operations; in particular, AVCAL items would probably not be included in an ABDR spares stockpile. In the following analysis, "damage" refers to the need to replace a candidate part on the aircraft. Battle damage not leading to a demand for a replacement part is treated as "no damage" by these models. In particular, damage requiring the replacement of "noncandidate" parts is treated as "no damage" by the spares models.

The DBS-Part model is a direct adaptation of the demand-based Aviation Supply Office (ASO) spares model currently used to develop AVCALs for most aircraft types. The stock level of a part under this rule depends only on the predicted damage rate for the part. Assuming damage incidents are described by a binomial distribution, just enough parts are stocked to ensure that the probability of a stockout is no more than 10 percent.

The RBS-Part model is analogous to the readiness-based spares model used by ASO to develop AVCAL lists for the F-18. The stock level of a part depends on the unit cost of the part as well as the predicted damage rate of the part. Essentially, the part stockpile is built part by part, where at each step, one unit of the item offering the largest gain in "readiness" per unit cost is added. The process is stopped when a specified cost goal is reached. In this analysis, when compared to the DBS models, the total cost for the stockpile produced by RBS-Part was constrained to be just below the cost of the corresponding DBS-Part stockpile. In other parts of the analysis, other cost goals were used; in each case, the cost goal used is clearly indicated in this report.

[1] DBS stands for demand-based sparing, and RBS stands for readiness-based sparing. DBS-P, RBS-P, DBS-DI, and RBS-DI will be used to denote the DBS part, RBS part, DBS damage incident, and RBS damage incident models, respectively.

Both of the spares models described above assume independence between part damage rates. As discussed earlier, this assumption does not seem reasonable when applied to battle damage. The "damage incident" models are designed to replace the unacceptable assumption of independence between part damage with the more acceptable assumption of independence between damage incidents.

The DBS-Damage Incident model is a direct adaptation of the DBS-Part model, where damage incidents play the role of "parts" in the DBS-Part model. In effect, "bundles" of parts, as defined by the dependence among parts in battle damage incidents, would be carried in the ABDR part stockpile. For example, suppose there are two damage incidents: incident A requires parts 1 and 2, while incident B requires parts 2 and 3 for repairs. Suppose the probability of experiencing more than four of incident A is less than 10 percent, and experiencing more than three of incident B is less than 10 percent. Then the "damage incident stockpile" is four of incident A and three of incident B. In terms of parts, this corresponds to the following stockpile: four of part 1, seven of part 2, and three of part 3.

In an analogous manner, the RBS-Damage Incident model is adapted from the RBS-Part model.

COMPARISON OF THE ALTERNATIVE SPARES MODELS

This analysis is based on the premise that predicted damage rates and actual damage rates are likely to be different, and a spares model offering flexibility (e.g., model B in figure 1) is preferable to a spares model offering narrow optimality (e.g., model A in figure 1). The flexibility of the four alternative spares models was investigated over a range of three possible relationships between predicted and actual damage rates:

- Full confidence: Actual damage rates are the same as the predicted damage rates.
- No confidence: No relationship or correlation exists between predicted and actual damage rates.
- Some confidence: Predicted rates are close to the actual damage rates.

A precise definition of the "confidence factor" parameter used in this analysis is given in the technical section of this paper.

The models were compared in terms of a random variable CANN, roughly corresponding to the number of cannibalized aircraft needed to meet the deficiencies in stockpiles of ABDR spares. A simple illustration of the way CANN is calculated is given in table 1.

TABLE 1

EXAMPLES OF CALCULATING CANN

	Part 1		Part 2		
Example number	Stocked	Demanded	Stocked	Demanded	CANN
1	3	5	7	6	2
2	3	2	7	5	0
3	3	7	7	12	5

Initial work was based on a list of 23 major structural assemblies for the F-14 that the Naval Air Systems Command (NAVAIR) is considering as candidates for ABDR spares stockpiles. A set of 30 damage incidents was hypothesized, and the performance of the four spares models in this situation was analyzed. At later stages of the analysis, more general situations were considered in which both the cost of ABDR spares and the list of parts associated with each damage incident were described by probability distribution functions. The generalized cost distribution used was derived from cost data for major aircraft assemblies provided by NAVAIR. The generalized distributions of part-to-damage incidents were designed to allow the parameterization of the level of dependence between part damage rates.

The numbers of parts (23) and damage incidents (30) used in the initial work on F-14 major assemblies were retained throughout the analysis. Because of computer limitations, it was not feasible to extend the analysis to substantially larger numbers of parts and damage incidents. This was not judged to severely limit the generality of the analysis, because a wide range of damage rates and distributions of part-to-damage incidents were considered in the comparison of the spares models.

A sample of results is contained in figure 2, and tables 2 and 3.[1] Figure 2 compares the probability distributions of CANN. Tables 2 and 3 show a more direct comparison between the spares models. Model A is said to be better than model B if model A leads to fewer "cannibalized aircraft" than model B; that is, if $\text{CANN}_B - \text{CANN}_A > 0$. These sample results correspond to a stockpile of major aircraft assemblies and are based on a support period of 3,000 sorties without resupply, an overall damage rate of approximately 0.008 per sortie, and "some confidence" in the predicted damage rates. In figure 2 and table 2, the cost of the RBS stockpile is limited to the cost of the corresponding DBS stockpile; in this comparison, the cost of a Damage Incident stockpile is higher than the cost of the Part stockpile. Table 3 shows an equal-cost comparison between the RBS models.

TABLE 2

SAMPLE RESULTS: COMPARISON OF FOUR SPARES MODELS

The "probability Model A performs better than Model B" is $\text{Prob}\{\text{CANN}_B - \text{CANN}_A > 0\}$.

Parameters used in this comparison
RBS-P cost goal: DBS-P cost
RBS-DI cost goal: DBS-DI cost
Some confidence in predicted damage rates

DBS-P versus RBS-P		DBS-P versus DBS-DI	
Prob. DBS-P is better:	0.11	Prob. DBS-P is better:	0.01
Prob. RBS-P is better:	0.41	Prob. DBS-DI is better:	0.29
RBS-P versus DBS-DI		**RBS-P versus DBS-DI**	
Prob. DBS-P is better:	0.06	Prob. RBS-P is better:	0.28
Prob. RBS-DI is better:	0.48	Prob. DBS-DI is better:	0.21
DBS-P versus RBS-DI		**RBS-DI versus RBS-DI**	
Prob. RBS-P is better:	0.06	Prob. DBS-DI is better:	0.12
Prob. RBS-DI is better:	0.24	Prob. RBS-DI is better:	0.34

[1] Tables 2 and 3 are taken from "series 5" of appendix A.

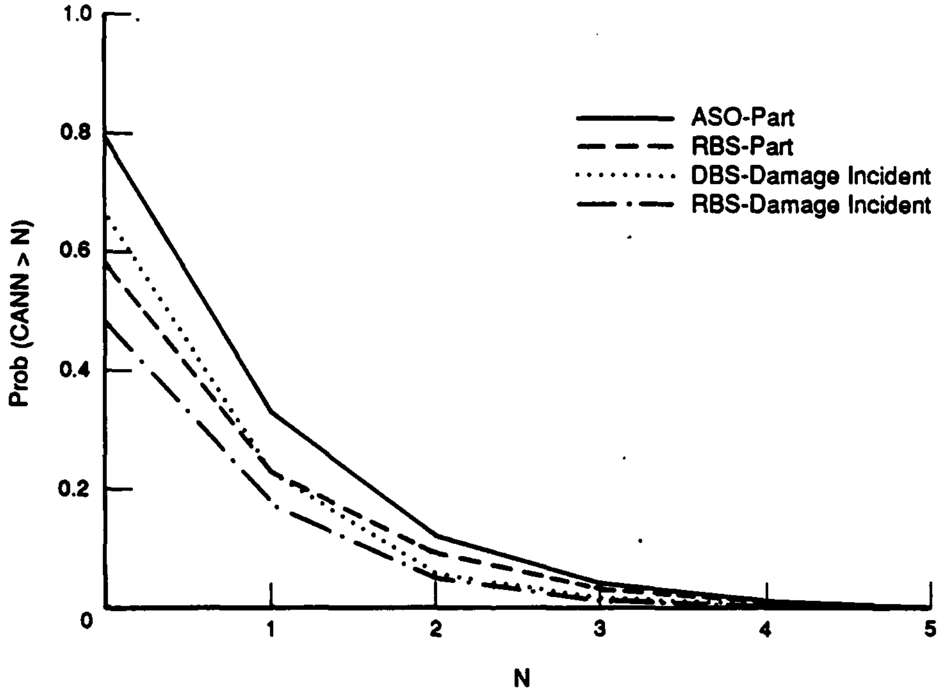

FIG. 2: SAMPLE RESULTS: PROBABILITY-DISTRIBUTION FUNCTIONS OF CANN

TABLE 3

SAMPLE RESULTS: COMPARISON OF EQUAL-COST GOALS OF RBS MODELS

The "probability Model A performs better than Model B" is Prob$\{\text{CANN}_B - \text{CANN}_A > 0\}$.

Parameters used in this comparison
RBS-P cost goal: slightly higher than DBS-DI cost
RBS-DI cost goal: slightly higher than DBS-DI cost
Some confidence in predicted damage rates

RBS-P versus RBS-DI	
Prob. RBS-P is better:	0.23
Prob. RBS-DI is better:	0.05

The relationship between the RBS and DBS models, illustrated in figure 2 and table 2, held consistently among the situations studied. The RBS (Part or Damage Incident) approach was consistently better over all three levels of confidence in predicted battle damage rates than the corresponding DBS approach. The results indicate that the readiness-based approach produces a better spares stockpile than the demand-based approach.

Recall that in the comparison in table 2, the RBS-Part cost goal was set so that the cost of the RBS-Part stockpile was slightly below the cost of the DBS-Part stockpile. It is always true that the cost of the DBS-Damage Incident stockpile will be greater than the cost of the DBS-Part stockpile. Hence, in the table 2 comparison, the DBS-Damage Incident stockpile is higher in cost than the RBS-Part stockpile. The relationship between the DBS-Damage Incident and RBS-Part models illustrated in table 2 is surprising in view of this relative cost relationship. The RBS-Part model generally performed better than the DBS-Damage Incident model when predicted damage rates were not good predictions of actual damage rates; however, the opposite was true when the predicted damage rates were good predictions of the actual damage rates. Hence, the RBS-Part stockpile generally performed better than the more expensive DBS-Damage Incident stockpile. This result emphasizes the flexibility of the readiness-based approach.

Recall that in table 2, the RBS-Part and RBS-Damage Incident models had different cost goals, making it difficult to compare these two models. To differentiate their relative value, analysts made a number of equal-cost comparisons between the RBS-Part and RBS-Damage Incident model. Because of the high cost of some parts, even when the two models are run to equal-cost goals, there will probably be some (relatively small) difference in the cost of the stockpiles.

In the equal-cost comparison illustrated in table 3, the cost goals were set so that so that the RBS-Part stockpile cost would, on average, fall slightly below the cost of the RBS-Damage Incident model. In effect, this gives a slight advantage to the RBS-Damage Incident model. Table 3 contains typical results; the RBS-Part model generally performed better than the RBS-Damage Incident model in equal-cost comparisons. In fact, the RBS-Damage Incident had better performance only in a few cases: a subset of the comparisons in which the damage rates were high and there was little correlation between predicted and actual damage rates.

The excellent performance of the RBS-Part model was unexpected; it was anticipated that the independence assumption in the RBS-Part model would result in poor performance relative to the RBS-Damage Incident model. From the way the cost goals of table 3 were set, the RBS-Part stockpile costs slightly less on average; hence, the better performance is not due to a difference in cost. Relative to a DBS model, the RBS model will reduce the stock level of expensive items and use the savings to increase stock levels of less expensive parts. In this way, the probability of experiencing stockouts of inexpensive items is "greatly" reduced at the expense of "slightly increasing" the probability of stockouts for the expensive items. The RBS-Damage Incident model appears to carry this tendency of trading off expensive items for inexpensive items a bit too far. Relative to the RBS-Part model, the RBS-Damage Incident model appears to experience slightly more stockouts for expensive items and slightly fewer stockouts for inexpensive items. However, the RBS-Part model also does well in meeting demands for the inexpensive items. The overall effect is that the RBS-Part model generally performs slightly better than the RBS-Damage Incident model. A more detailed example is provided in appendix B.

Appendix A contains Prob$\{\mathrm{CANN}_A - \mathrm{CANN}_B > 0\}$, as in tables 2 and 3, for the different situations considered in the analysis. While graphs corresponding to figure 2 were also considered in the analysis, they did not add information and are not included in appendix A.

CONCLUSION

Among the four spares models considered in this analysis, the RBS models provide better performance than the corresponding DBS models. When RBS-Part and RBS-Damage Incident are used to equal-cost goals, the RBS-Part model performs better overall. In addition, RBS-Part is an easier model to use because damage rates and similar data are processed at the part level rather than the damage-incident level. The analysis indicates that the RBS-Part model should be adopted for determining ABDR material requirements.

TECHNICAL DETAILS

In this section, a number of technical details are given, including a discussion of the mathematics used in the analysis and precise definitions of the spares models and distribution functions used in the analysis.

MATHEMATICS

The underlying events in the mathematical formulation of the ABDR spares model problem are damage incidents. In terms of the NWC models used to study aircraft vulnerability and susceptibility, each damage incident may correspond to a specific "shotline" in a vulnerable area model, or to a specific missile endgame model result. (See [1] for a discussion of these models.) In real-world terms, the set of damage incidents is the set of possible outcomes of a combat sortie. For mathematical convenience, the outcome in which no parts are required, either because no damage is experienced or none of the candidate parts are needed for repairs, will always be included in the set of damage incidents. One assumption made with respect to damage incidents is that the damage incident experienced on each sortie is independent of the damage incidents experienced on other sorties.

Assume there is a set of n_p parts that are candidates for an ABDR stockpile, and assume that there is a set of n_d distinct damage incidents. The following notation is used throughout this section to denote ABDR-related factors over some period of time T encompassing N combat sorties:

- The vectors $\vec{0}$ and $\vec{1}$ denote $(0,0,\cdots,0)$ and $(1,1,\cdots,1)$, respectively. Vector addition, scalar multiplication, and "dot product" (denoted $\cdot$) have the usual meaning. Two vectors satisfy $\vec{v} \leq \vec{w}$ if the inequality $(\vec{v})_i \leq (\vec{w})_i$ holds for the i components of the vectors.
- The vector $\vec{r_p}$ describes predicted damage rates. The i-th component denotes the probability that damage incident i will be experienced on a sortie. Recall that the no-damage incident is included with appropriate r_p and note that $\vec{r_p} \cdot \vec{1} = 1$.

- The vector $\vec{r_a}$ describes actual damage rates. These are damage rates that will be experienced in combat and may be different from the predicted battle damage rates.

- The vector $\vec{k}$ describes the number of each type of damage incident experienced over a period of time T. The i-th component $(\vec{k})_i$ denotes the number of times damage incident i was experienced. The probability distribution of $\vec{k}$ after N combat sorties is the multinomial distribution:

$$\text{Prob}\{\vec{k} = \vec{x} \text{ after } N \text{ sorties}\} = M(\vec{x}, \vec{r_a}, N)$$

$$\binom{N}{(\vec{x})_1, \cdots, (\vec{x})_{n_d}} (\vec{r_a})_1^{(\vec{x})_1} \cdots (\vec{r_a})_{n_d}^{(\vec{x})_{n_d}}.$$

- The vector $\vec{D}_i$ specifies the list of parts required to repair damage incident i. For example, if $\vec{D}_i = (1, 0, 1, \cdots)$, then one of part number 1, one of part number 3, and so on, are needed to repair damage incident i. This analysis assumes that $\vec{D}_i \leq \vec{1}$

- The vector $\vec{d}$ describes the number of each type of part demanded over the time period T. Given the vector $\vec{k}$ describing the damage incidents experienced this period, $(\vec{d})_i = \sum_{j=1}^{n_d} (\vec{D}_j)_i (\vec{k})_j$. Note that $\vec{d}$ and $\vec{r_a}$ are dependent random vectors; and $\vec{d}$ also depends on the number of combat sorties N, and the part-to-damage incident vectors $\vec{D}$.

- The vector $\vec{s}$ describes stock levels. The i-th component $(\vec{s})_i$ denotes the stock level of item i in the ABDR stockpile. Similarly, $\vec{c}$ describes the costs of the ABDR parts. The stock level produced by a spares model is a function of the predicted damage rates, the number of combat sorties N to be supported by the stockpile, cost, and the part-to-damage-incident vectors $\vec{D}$; in functional notation, $\vec{s} = \vec{s}(\vec{r_p}, N, \vec{c}, \vec{D})$.

Measure of Effectiveness Used To Compare Spares Models

The models for spare parts requirements were compared in terms of a random variable CANN, computed by comparing the list of parts demanded to the list of parts stocked. Let $(\vec{s})_i$ be the stock level of part i, and let $(\vec{d})_i$ be the demand for part i. Assuming no resupply, $\max\{0, (\vec{d})_i - (\vec{s})_i\}$ is the number of stockouts for part i. The random variable

$$\text{CANN} = \max\{\max\{0, (\vec{d})_i - (\vec{s})_i\} \mid \text{all } i\}$$

is analogous to the number of cannibalized aircraft required to fill the stockouts. The notations CANN_A and CANN_B are used to represent CANN under the sparing models A and B, respectively. The random variable $\text{CANN}_A - \text{CANN}_B$ is used to directly compare spares model A with spares model B.

It is convenient to think of CANN as a function of the stock level, demand for parts, predicted and actual damage rates, the number of combat sorties N, cost, and the part-to-damage-incident vectors $\vec{D}$; in functional notation, $\text{CANN} = \text{CANN}(\vec{s}, \vec{d}, \vec{r_a}, \vec{r_p}, N, \vec{c}, \vec{D})$. Also, it is convenient to think of the variables of interest as belonging to a probability space $X = \{(\vec{s}, \vec{d}, \vec{r_a}, \vec{r_p}, N, \vec{c}, \vec{D})\}$ equipped with a probability measure μ. In this space, N, $\vec{c}$, and $\vec{D}$ are all independent of other variables; $\vec{s}$ is completely determined by $\vec{r_p}$, $\vec{c}$, and N; and there are dependencies between $\vec{d}$, $\vec{r_a}$, and $\vec{r_p}$. This allows certain quantities of interest in the analysis, in particular $\text{Prob}\{\text{CANN}_A - \text{CANN}_B > 0\}$, to be represented as an integral

$$I = \int_X f d\mu$$

for some specific function f. If f is defined by

$$f(\vec{s}, \vec{d}, \vec{r_a}, \vec{r_p}, N, \vec{c}, \vec{D}) = \begin{cases} 1 & \text{if } \text{CANN}(\cdot)_A - \text{CANN}_B(\cdot) > 0 \\ 0 & \text{otherwise} \end{cases},$$

then $I = \text{Prob}\{\text{CANN}_A - \text{CANN}_B > 0\}$.

The integral I introduced above is difficult to calculate because the number of numerical calculations needed to compute the exact value of the integral grows rapidly with the number of damage incidents and parts considered. Stochastic integration methods were selected as the easiest way of estimating the integral. Justification of the techniques used are given below. A good general reference on stochastic integration is [2].

Estimates of the quantities $\text{Prob}\{\text{CANN}_A - \text{CANN}_B > 0\}$ were made over a variety of damage rates, costs, and part-to-damage-incident tables, each combination giving rise to a distinct probability measure μ. The parameters and distribution functions used to describe damage rates, costs, and part-to-damage-incident tables are given later in this section. The estimates are reported in appendix A.

Stochastic Integration

The integrals considered in this analysis are of the form $I = \int_X f d\mu$, where $f : X \rightarrow \{0,1\}$ and μ is a probability measure on X. The statistic θ_n is defined by $\theta_n = \frac{1}{n}\sum_{i=1}^{n} f(x_i)$, where $\{x_i : i = 1, \cdots, n\}$ is an independent random sample of size n from X.

The expected value of the statistic θ_n is

$$E(\theta_n) = \int_X f d\mu = I \quad ,$$

and the variance of the statistic is

$$V(\theta_n) = \frac{1}{n}(\int_X f^2 d\mu - I^2) \quad .$$

By Chebyshev's inequality, and using the facts that $I - I^2 \leq 1/4$ on $[0,1]$ and $f^2 = f$,

$$\text{Prob}\{|\theta_n - I| \geq \epsilon\} \leq \frac{1}{\epsilon^2} V(\theta_n)$$

$$= \frac{1}{\epsilon^2 n}(\int_X f^2 d\mu - I^2)$$

$$= \frac{1}{\epsilon^2 n}(I - I^2) \leq \frac{1}{4\epsilon^2 n} \quad .$$

This allows one to determine the number of samples required to give a specified level of accuracy to the estimate θ_n. The sample size used in all estimates reported in this analysis was $n = 500$, ensuring for example that

$$\text{Prob}\{|\theta_n - I| \geq 0.1\} \leq 0.05;$$

that is, 95 percent confidence that the estimate is within 0.1 of the true value.

Numerical Calculations

Two separate computer programs were written to carry out the stochastic integration technique described above. The first program generates 500 random samples from the probability space X. For each of the random samples, four values of CANN, one for each of the four spares models, are calculated and written to an intermediate file. A listing of this program, corresponding to the most general definition of the probability space X considered, is included in appendix C. The second computer program reads values of CANN from the intermediate file, then computes and averages the corresponding values of the function f.

An outline of the procedure used to generate a single random sample from the probability space $X = \{(\vec{s}, \vec{d}, \vec{r_a}, \vec{r_p}, N, \vec{c}, \vec{D})\}$ is given below:

- N: In all of the situations considered, the number of sorties N was held constant.
- $\vec{c}$: In some situations considered, the cost vector $\vec{c}$ was constant. In other situations, a random vector $\vec{c}$ was drawn from the (independent) probability distribution for cost.
- $\vec{D}$: In some situations considered, the matrix describing the list of parts needed to repair a damage incident was constant. In other situations, a random matrix $\vec{D}$ was drawn from the (independent) probability distribution describing part-to-damage-incident relationships.
- $\vec{r_p}$: A random vector describing the predicted damage rates was drawn from the probability distribution for $\vec{r_p}$.
- $\vec{r_a}$: The predicted and actual damage rates are not necessarily independent. After a specific predicted damage rate vector $\vec{r_p}$ was drawn, an actual damage rate vector $\vec{r_a}$ was drawn from the conditional distribution function of $\vec{r_a}$ given $\vec{r_p}$.
- $\vec{d}$: The random vector $\vec{k}$ describing damage-incident experience is dependent on the actual damage rate $\vec{r_a}$. After a specific actual damage-rate vector $\vec{r_a}$ was drawn, a damage-experience vector $\vec{k}$ was drawn from the conditional distribution function of $\vec{k}$, given $\vec{r_a}$ and N. The list of parts required, $\vec{d}$, was then computed directly from $\vec{k}$ using the part-to-damage-incident matrix $\vec{D}$.

- $\vec{s}$: The vector of stock levels $\vec{s}$ was computed directly from the cost vector $\vec{c}$, N, and the vector of predicted damage rates $\vec{r_p}$ using the appropriate spares model algorithm.

DEFINITION OF SPARES MODELS

Four different spares models were evaluated; each model is defined below.

In the following,

$$B(k,p,n) = \sum_{j=0}^{k} \binom{n}{j} p^j (1-p)^{n-j}$$

denotes the binomial distribution function. Each model assumes that the ABDR spares are to cover demands over a fixed period of time, that is, a fixed number of combat sorties, with no resupply or repair of damaged parts.

The <u>DBS-Part</u> model is a direct adaptation of the supply-oriented model traditionally used in developing AVCALs. The stock level of part i is the first integer k satisfying $B(k, r_i, N) > 0.9$, where $r_i = \sum_{j=1}^{n_d} (\vec{D_j})_i (\vec{r_p})_j$ is the predicted damage rate for part i and N is the number of combat sorties. Note that the stock levels computed for each part are independent of the stock levels computed for other parts.

The <u>RBS-Part</u> model is analogous to the readiness-based spares model used by ASO. Given a stock level $\vec{s}$, the probability that there are no stockouts over a period of N combat sorties is taken as a measure of the readiness provided by the stockpile. Because independence is assumed in this model, the probability of no stockouts may be written

$$\prod_{j=1}^{n_p} B((\vec{s})_j, r_j, N) \quad .$$

An inductive procedure is used to build a stock level from a starting position $\vec{s_0} = \vec{0}$. At each step in the process, one unit of the item offering the largest gain in "readiness" per unit cost is added to the stockpile. More precisely, let $\vec{s_n}$ describe the stock level after step n. Pick M to satisfy

$$\frac{1}{(\vec{c})_k}\left(\prod_{j=1}^{n_p} B((\vec{s_n})_j + \delta_{jk}, r_j, N) - \prod_{j=1}^{n_p} B((\vec{s_n})_j, r_j, N)\right)$$

$$\leq$$

$$\frac{1}{(\vec{c})_M}\left(\prod_{j=1}^{n_p} B((\vec{s_n})_j + \delta_{jM}, r_j, N) - \prod_{j=1}^{n_p} B((\vec{s_n})_j, r_j, N)\right)$$

for all $k = 1, \cdots, n_p$ where

$$\delta_{ij} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{otherwise} \end{cases} .$$

Let $\vec{s_{n+1}} = \vec{s_n} + (\delta_{1M}, \cdots, \delta_{n_pM})$. The process is halted when a specified cost goal is reached.

Both of the above spares models assume independence between part "failures." This assumption does not seem reasonable when applied to battle damage. The raw output of the NWC/NADC modeling effort will in effect be stated in terms of damage incidents and must be condensed into part-level damage rates. Moreover, it seems more reasonable to assume the independence of damage incidents than the independence of part damage. This motivates the following alternative spares models.

The DBS-Damage Incident model is a dirrct adaptation of the DBS-Part model, where damage incidents rather than demands for parts are monitored. In other words, each damage incident corresponds to a "bundle" of parts needed to repair that particular damage. Enough bundles of parts are stocked to ensure that demands for that particular bundle are met with at least a probability of 90 percent. More precisely, the "stock" level of damage-incident i is the first integer k where $B(k, (\vec{r_p})_i, N) > 0.9$. The damage-incident "stock levels" $\vec{S_d}$ are converted to part stock levels $\vec{s}$ via $\vec{s} = \sum_{i=1}^{n_d} (\vec{S_d})_i \vec{D_i}$.

In an analogous manner, the RBS-Damage Incident model is adapted from the RBS-Part model.

DAMAGE RATE, COST, AND PART-TO-DAMAGE-INCIDENT DISTRIBUTIONS

This section defines the damage-rate distributions, part-to-damage-incident distributions, and cost distributions used in the analysis.

F-14 Assemblies

Initial work began with a list of 23 candidate major structural assemblies for ABDR on the F-14 and a hypothetical distribution of the assemblies among 30 distinct damage incidents. The list of assemblies and costs, listed in table 4, were provided by NAVAIR. The part-to-damage-incident table is given in table 5.

Generalized Part-to-Damage-Incident Distributions

Computer limitations would not allow the consideration of substantially higher numbers of parts and damage incidents; hence, 23 parts and 30 damage incidents were used throughout the analysis. However, it was relatively easy to consider more general part-to-damage-incident distributions than the hypothetical F-14 assemblies distributions shown in table 5.

Given a fixed number $r \in (0,1)$, the following procedure was used to generate a part-to-damage-incident matrix $\vec{D}$. For each fixed damage incident, a random number was drawn for each part j to determine if part j was needed to repair the damage incident. The probability that part j was needed was exactly r. If, at the end of this process, at least one part was needed to repair the damage incident, the list of parts just selected was accepted. If no parts were selected, the (empty) list of parts was rejected and the selection process was repeated.

The probability distribution function corresponding to the procedure above is given by the following function:

$$\text{Prob}\{\text{number of parts included} \geq x\}$$

$$= F(r,x) = \frac{1 - B(x-1, r, 23)}{1 - B(0, r, 23)} \quad .$$

TABLE 4

MAJOR ASSEMBLIES AND COSTS FOR CANDIDATE F-14

Major assembly	Cost (thousand $)
Outer wing panel (left)	617
Outer wing panel (right)	617
Aileron (left)	780
Aileron (right)	780
Leading edge flap (left)	28
Leading edge flap (right)	28
Trailing edge flap (left)	21
Trailing edge flap (right)	21
Horizontal stabilizer (left)	445
Horizontal stabilizer (right)	445
Vertical tail (left)	106
Vertical tail (right)	132
Rudder (left)	42
Rudder (right)	42
Main landing gear (left)	43
Main landing gear (right)	109
Engine inlet assembly (left)	30
Engine inlet assembly (right)	30
Canopy	143
Windscreen	28
Nose landing gear	92
Afterburner assembly	11
Radome	35

TABLE 5

HYPOTHETICAL F-14 DAMAGE INCIDENTS

Damage incident number	Parts requiring replacement
1-23	One for each of the 23 candidate parts
24	Outer wing panel (left) Aileron (left) Leading edge flap (left) Trailing edge flap (left)
25	Outer wing panel (right) Aileron (right) Leading edge flap (right) Trailing edge flap (right)
26	Canopy Windscreen
27	Main landing gear (left) Engine inlet assembly (left)
28	Main landing gear (right) Engine inlet assembly (right)
29	Horizontal stabilizer (left) Vertical tail (left) Rudder (left) Afterburner assembly
30	Horizontal stabilizer (right) Vertical tail (right) Rudder (right) Afterburner assembly

Each of the 23 candidate parts has an equal probability of being one of the x parts selected for a damage incident. Three different values of the parameter r are used in the analysis. The parameter value $r = 0.023$ was selected to match with the hypothetical F-14 assemblies distribution shown in table 5: 23 percent of the damage incidents include two or more parts. The parameter value $r = 0.11$ was selected so the probability of having more than two parts included in a damage incident was 50 percent. The parameter value $r = 0.2$ was selected so the probability of having more than four parts included in a damage incident was 50 percent. On average, each part is included in 1.6 damage incidents when $r = 0.023$, 3.5 damage incidents when $r = 0.11$, and 6.0 damage incidents when $r = 0.2$.

Generalized Cost Distribution

The following distribution of cost was used to generalize the cost distribution:

$$\text{Prob}\left\{\frac{C}{C_{max}} \le t\right\}$$

$$= \begin{cases} 4t & 0 \le t \le 0.2 \\ 0.8 & 0.2 \le t \le 0.5 \\ 0.4t + 0.6 & 0.5 \le t \le 1 \end{cases} .$$

This distribution was chosen to match reasonably well with the distributions of major aircraft assembly costs provided by NAVAIR. Figure 3 compares the cost distribution function with the actual distribution of major aircraft assembly costs for the A-6, F-14, and F-18.

Damage-Rate Distributions

A two-parameter distribution of predicted and actual damage rates was used in this analysis. The "maximum DI rate" parameter S defined the range of possible damage rates; the "confidence factor" C defined the relationship between

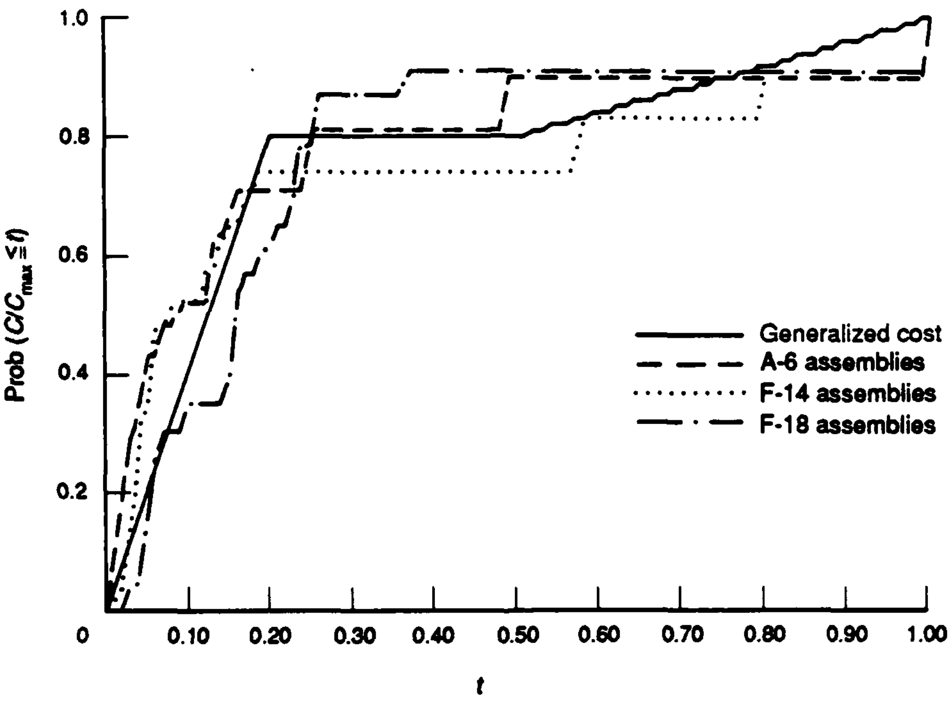

FIG. 3: GRAPH OF COST-DISTRIBUTION FUNCTION

the predicted and actual damage rates. Precisely, the range of predicted (r_p) and actual (r_a) damage rates for each of the 30 damage incidents actually requiring a part was uniformly distributed in the set

$$\{(r_p, r_a) : 0 \leq r_p \leq S \text{ and } Cr_p \leq r_a \leq Cr_p + S(1 - C)\} \ .$$

A parameter value $C = 0$ corresponds to "no confidence," a value $C = 0.5$ corresponds to "some confidence," and a value $C = 1$ corresponds to "full confidence."

Figure 4 displays pictorially the effect of the confidence-factor parameter and the maximum DI rate parameter. The maximum DI rate parameter S sets the maximum allowable damage rate. For example, if $S = 0.005$, the probability of experiencing a specific damage incident on a sortie is no more than 0.005. Note that $C = 1$ (full confidence) forces the predicted and actual rates to be equal, and that the predicted and actual rates are unrelated when $C = 0$ (no confidence). When $C = 0.5$ (some confidence), the actual damage rate and predicted damage rate both fall within an interval of width $\frac{S}{2}$.

Step 1: Pick predicted damage rate r_p in interval [0, S].

Step 2: Pick actual damage rate r_a. The range of permitted values for r_a is shaded below.

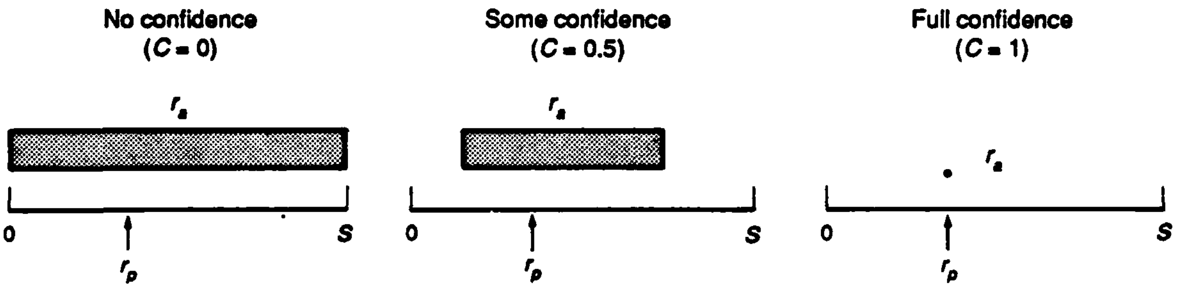

FIG. 4: EFFECT OF THE CONFIDENCE-FACTOR PARAMETER

REFERENCES

[1] Robert E. Ball. The Fundamentals of Aircraft Combat Survivability Analysis and Design. New York: American Institute of Aeronautics and Astronautics, 1985

[2] Reuven Y. Rubinstein. Simulation and the Monte Carlo Method. New York: John Wiley & Sons, 1981

APPENDIX A
DETAILED RESULTS

APPENDIX A

DETAILED RESULTS

This appendix contains comparisons of the four ABDR spares requirements models over a wide range of parameter and distribution values. A total of eight different combinations of maximum DI rate, part-to-DI distribution, and cost distribution were considered in the analysis, as listed in table A-1. The term "series," defined by this table, is used to refer to these combinations. Each of the tables that follow compares two of the models in terms of the statistic "probability Model A performs better than Model B." More precisely, this statistic is Prob$\{\mathrm{CANN}_B - \mathrm{CANN}_A > 0\}$.

The relative costs of the stockpiles associated with the spares models must be taken into account when comparing the spares models. The cost of the DBS-Damage Incident stockpile is always higher than the cost of the corresponding DBS-Part stockpile. The cost associated with the RBS stockpile is indicated at the top of each table. For most of the comparisons, the cost of the RBS stockpile was forced to remain slightly below the cost goal: the RBS algorithm was halted when the addition of the next part would exceed the cost goal; this last part *was not added* to the stockpile. This rule was used in all of the following comparisons except the equal-cost comparisons of the RBS-Part and RBS-Damage Incident models (tables A-8 and A-9).

In the equal-cost comparisons between the RBS-Part and RBS-Damage Incident models, the cost of the RBS stockpiles was allowed to exceed the cost goal slightly: the RBS algorithm was halted when the addition of the next part would exceed the cost goal; this last part *was added* to the stockpile. This case is identified in the following tables as: "RBS cost is slightly higher than the cost goal." The "slightly higher" rule causes the RBS-Damage Incident stockpile cost to be slightly higher than the RBS-Part cost on average. The other rule causes the RBS-Part stockpile cost to exceed the RBS-Damage Incident cost on average. The "slightly higher" rule was selected for the equal-cost comparisons of the RBS models because it gives the RBS-Damage Incident model a slight advantage.

TABLE A-1

SERIES COMPARED

Series	Max. DI rate	Part-to-DI dist.	Cost dist.
1	0.0005	F14 Assem.	F14 Assem.
2	0.005	F14 Assem.	F14 Assem.
3	0.0005	F14 Assem.	Genl.
4	0.005	F14 Assem.	Genl.
5	0.0005	Genl., r = 0.023	Genl.
6	0.005	Genl., r = 0.023	Genl.
7	0.0005	Genl., r = 0.11	Genl.
8	0.0005	Genl., r = 0.2	Genl.
9	0.005	Genl., r = 0.2	Genl.

Tables A-2 thr·ugh A-9 contain comparisons based on 23 parts distributed among 30 damage incidents and a support period of 3,000 sorties. Discussions and definitions of the parameter values and distributions are contained in the technical section of this paper. The accuracy or measurement error of these statistics is also discussed in the technical section of this paper.

TABLE A-2

DBS-P VERSUS RBS-P
(Equal cost)

RBS cost goal for this comparison
RBS-P cost: DBS-P

Series	Conf. factor	Prob. DBS-P is better	Prob. RBS-P is better
1	No	0.08	0.53
	Some	0.09	0.43
	Full	0.11	0.38
2	No	0.08	0.74
	Some	0.12	0.72
	Full	0.20	0.47
3	No	0.12	0.44
	Some	0.13	0.38
	Full	0.13	0.33
4	No	0.12	0.76
	Some	0.16	0.68
	Full	0.26	0.47
5	No	0.11	0.46
	Some	0.11	0.41
	Full	0.15	0.32
6	No	0.13	0.75
	Some	0.15	0.70
	Full	0.24	0.46
7	No	0.14	0.49
	Some	0.17	0.38
	Full	0.21	0.32
8	No	0.14	0.46
	Some	0.21	0.36
	Full	0.19	0.34

TABLE A-3

DBS-P VERSUS DBS-DI
(DBS-P cost $\leq$ DBS-DI cost)

RBS cost goal for this comparison
Not applicable

Series	Conf. factor	Prob. DBS-P is better	Prob. DBS-DI is better
1	No	0.02	0.34
	Some	0.01	0.34
	Full	0.01	0.40
2	No	0.01	0.76
	Some	0.00	0.73
	Full	0.00	0.65
3	No	0.02	0.30
	Some	0.02	0.33
	Full	0.01	0.36
4	No	0.01	0.77
	Some	0.00	0.76
	Full	0.00	0.66
5	No	0.01	0.33
	Some	0.01	0.29
	Full	0.00	0.32
6	No	0.00	0.71
	Some	0.00	0.61
	Full	0.00	0.49
7	No	0.01	0.58
	Some	0.01	0.51
	Full	0.00	0.51
8	No	0.00	0.69
	Some	0.00	0.62
	Full	0.00	0.57

TABLE A-4

DBS-P VERSUS RBS-DI
(DBS-P cost $\leq$ RBS-DI cost)

RBS cost goal for this comparison
RBS-DI cost: DBS-DI

Series	Conf. factor	Prob. DBS-P is better	Prob. RBS-DI is better
1	No	0.07	0.60
	Some	0.04	0.52
	Full	0.04	0.47
2	No	0.02	0.89
	Some	0.04	0.86
	Full	0.03	0.65
3	No	0.05	0.55
	Some	0.06	0.51
	Full	0.06	0.47
4	No	0.03	0.88
	Some	0.05	0.83
	Full	0.06	0.65
5	No	0.05	0.53
	Some	0.06	0.48
	Full	0.09	0.40
6	No	0.04	0.87
	Some	0.06	0.80
	Full	0.12	0.55
7	No	0.03	0.66
	Some	0.03	0.54
	Full	0.02	0.51
8	No	0.01	0.71
	Some	0.00	0.62
	Full	0.00	0.57

TABLE A-5

RBS-P VERSUS DBS-DI
(RBS-P cost ≤ DBS-DI cost)

RBS cost goal for this comparison
RBS-P cost: DBS-P

Series	Conf. factor	Prob. RBS-P is better	Prob. DBS-DI is better
1	No	0.40	0.18
	Some	0.29	0.22
	Full	0.20	0.23
2	No	0.67	0.25
	Some	0.58	0.25
	Full	0.26	0.38
3	No	0.29	0.20
	Some	0.24	0.25
	Full	0.16	0.24
4	No	0.60	0.23
	Some	0.49	0.30
	Full	0.20	0.40
5	No	0.29	0.21
	Some	0.28	0.21
	Full	0.17	0.25
6	No	0.49	0.31
	Some	0.46	0.32
	Full	0.23	0.39
7	No	0.18	0.36
	Some	0.11	0.40
	Full	0.06	0.40
8	No	0.06	0.49
	Some	0.03	0.49
	Full	0.02	0.45

TABLE A-6

RBS-P VERSUS RBS-DI
(RBS-P cost $\leq$ RBS-DI cost)

RBS cost goal for this comparison
RBS-P cost: DBS-P
RBS-DI cost: DBS-DI

Series	Conf. factor	Prob. RBS-P is better	Prob. RBS-DI is better
1	No	0.08	0.27
	Some	0.06	0.27
	Full	0.02	0.26
2	No	0.09	0.66
	Some	0.08	0.61
	Full	0.02	0.51
3	No	0.06	0.31
	Some	0.04	0.33
	Full	0.03	0.29
4	No	0.08	0.67
	Some	0.06	0.66
	Full	0.02	0.52
5	No	0.06	0.29
	Some	0.06	0.24
	Full	0.05	0.19
6	No	0.08	0.62
	Some	0.14	0.50
	Full	0.10	0.37
7	No	0.08	0.45
	Some	0.06	0.42
	Full	0.04	0.40
8	No	0.03	0.53
	Some	0.01	0.51
	Full	0.01	0.45

TABLE A-7

DBS-DI VERSUS RBS-DI
(Equal cost)

RBS cost goal for this comparison
RBS-DI cost: DBS-DI

Series	Conf. factor	Prob. DBS-DI is better	Prob. RBS-DI is better
1	No	0.14	0.50
	Some	0.13	0.37
	Full	0.13	0.28
2	No	0.13	0.75
	Some	0.12	0.70
	Full	0.17	0.36
3	No	0.11	0.38
	Some	0.11	0.34
	Full	0.13	0.27
4	No	0.12	0.73
	Some	0.15	0.68
	Full	0.18	0.32
5	No	0.11	0.37
	Some	0.12	0.34
	Full	0.19	0.22
6	No	0.12	0.72
	Some	0.16	0.61
	Full	0.24	0.30
7	No	0.13	0.29
	Some	0.11	0.18
	Full	0.10	0.10
8	No	0.10	0.18
	Some	0.09	0.10
	Full	0.05	0.06

TABLE A-8

RBS-P VERSUS RBS-DI
(Equal cost, "low" cost goal)

RBS cost goal for this comparison
RBS-P cost: slightly higher than DBS-P
RBS-DI cost: slightly higher than DBS-P

Series	Conf. factor	Prob. RBS-P is better	Prob. RBS-DI is better
1	No	0.21	0.10
	Some	0.17	0.09
	Full	0.13	0.10
2	No	0.33	0.30
	Some	0.28	0.21
	Full	0.22	0.14
3	No	0.17	0.09
	Some	0.17	0.10
	Full	0.13	0.09
4	No	0.30	0.28
	Some	0.26	0.26
	Full	0.23	0.16
5	No	0.20	0.12
	Some	0.25	0.09
	Full	0.24	0.09
6	No	0.31	0.42
	Some	0.38	0.30
	Full	0.40	0.14
7	No	0.41	0.09
	Some	0.32	0.11
	Full	0.31	0.10
8	No	0.39	0.11
	Some	0.38	0.10
	Full	0.31	0.15

TABLE A-9

RBS-P VERSUS RBS-DI
(Equal cost, "high" cost goal)

RBS cost goal for this comparison
RBS-P cost: slightly higher than DBS-DI
RBS-DI cost: slightly higher than DBS-DI

Series	Conf. factor	Prob. RBS-P is better	Prob. RBS-DI is better
1	No	0.15	0.10
	Some	0.14	0.06
	Full	0.08	0.05
2	No	0.33	0.29
	Some	0.29	0.24
	Full	0.12	0.08
3	No	0.18	0.07
	Some	0.14	0.06
	Full	0.10	0.06
4	No	0.27	0.28
	Some	0.27	0.27
	Full	0.15	0.11
5	No	0.25	0.08
	Some	0.23	0.05
	Full	0.18	0.05
6	No	0.33	0.42
	Some	0.46	0.22
	Full	0.32	0.09
7	No	0.31	0.08
	Some	0.25	0.05
	Full	0.14	0.03
8	No	0.23	0.02
	Some	0.15	0.02
	Full	0.07	0.01

APPENDIX B

AN EXAMPLE COMPARING RBS-PART AND RBS-DAMAGE INCIDENT MODELS

APPENDIX B

AN EXAMPLE COMPARING RBS-PART AND RBS-DAMAGE INCIDENT MODELS

Table B-1 presents a more detailed example of the reasons for the excellent performance of the RBS-Part model in comparison to the RBS-Damage Incident model. The table contains estimates made under the "series 1" conditions of appendix A with some confidence in predicted damage rates. In particular, the F-14 assemblies cost and part-to-damage-incident matrix was used. These parameters, taken from tables 4 and 5 of the main text, are repeated in columns 1 through 3 of table B-1. The other columns of table B-1 contain estimates of stockpile characteristics and performance:

- Columns 4 through 6 of table B-1 contain comparisons of the stock levels produced by the RBS-Damage Incident and RBS-Part models. For example, the probability that part 1 is stocked to a lower level under RBS-Damage Incident than RBS-Part was estimated to be 0.28. This is reported under part number 1 in the column headed "DI<P."
- Columns 7 through 10 contain the probability of certain types of stockout conditions. For example, the probability of a stockout of part 1 in the RBS-Part stockpile is 0.09. This is reported under part number 1 in the column labeled "P." The probability of a stockout for part 1 in the RBS-Damage Incident stockpile that would *not* have resulted in a stockout in the RBS-Part stockpile is 0.04. This is reported under part 1 in the column labeled "DI, not P."

The RBS models tend to trade expensive items for additional protection in inexpensive items. Intuitively, the RBS-Damage Incident model will find damage incidents 26 through 28 attractive because of the low cost and the increase in readiness produced by adding two parts to the stockpile simultaneously. Also intuitively, the RBS-Damage Incident model will not find parts 1 through 4 attractive because they are expensive both when considered separately and as part of damage incidents 24 through 25. By this intuitive argument, RBS-Damage Incident will tend to buy fewer of parts 1 through 4 and more of parts 15 through 20 than RBS-Part.

The intuitive argument above agrees with the estimates presented in table B-1. Note that the RBS-Damage Incident model tends to stock fewer of the expensive parts and parts included in expensive damage incidents (parts 1 through 8) than the RBS-Part model, and to stock more inexpensive parts and parts included in inexpensive damage incidents (particularly parts 15 through 20). Consequently, relative to the RBS-Part stockpile, the RBS-Damage Incident stockpile tends to experience more stockouts for expensive items and fewer stockouts for the inexpensive items.

The RBS-Damage Incident stockpile generally does better than the RBS-Part stockpile in meeting demands for inexpensive items. However, the RBS-Part stockpile already does quite well in meeting demands for inexpensive items. In relative terms, there is little difference in performance, with a probability of around 0.01 of a stockout under the RBS-Part model that would be covered under the RBS-Damage Incident model.

Both models have traded off expensive parts for additional coverage of inexpensive items. Their performance in meeting demands for expensive parts is relatively poor. However, the RBS-Part model performs better than the RBS-Damage Incident model in this performance category. In relative terms, the difference in performance is large, with up to a 0.04 probability of stockout under the RBS-Damage Incident model that would be covered by the RBS-Part model.

In summary, the RBS-Damage Incident model appears to carry the tendency of trading off expensive items for inexpensive items a bit too far. Relative to the RBS-Part model, the RBS-Damage Incident model appears to experience more stockouts for expensive items. While it is true that the RBS-Damage Incident model appears to experience fewer stockouts for inexpensive items, the RBS-Part model also does well in meeting demands for inexpensive items. The overall effect is that the RBS-Part model generally performs slightly better than the RBS-Damage Incident model.

TABLE B-1

DETAILED COMPARISON OF RBS-PART AND RBS-DAMAGE INCIDENT

		Cost	Comparison of stock levels (probability)			Probability of stockout			
DI	Part(s)	(T$)	DI<P	DI=P	DI>P	DI	P	DI, not P	P, not DI
1	1	617	0.28	0.66	0.06	0.12	0.09	0.04	0.00
2	2	617	0.27	0.68	0.05	0.14	0.11	0.04	0.01
3	3	780	0.20	0.73	0.07	0.12	0.11	0.03	0.01
4	4	780	0.21	0.71	0.08	0.10	0.08	0.04	0.01
5	5	28	0.62	0.37	0.01	0.04	0.01	0.02	0.00
6	6	28	0.61	0.39	0.01	0.02	0.01	0.01	0.00
7	7	21	0.66	0.34	0.01	0.03	0.01	0.02	0.00
8	8	21	0.68	0.32	0.00	0.02	0.01	0.01	0.00
9	9	445	0.02	0.59	0.39	0.06	0.07	0.00	0.02
10	10	445	0.04	0.65	0.31	0.06	0.08	0.00	0.02
11	11	106	0.16	0.62	0.22	0.02	0.01	0.01	0.00
12	12	132	0.12	0.70	0.19	0.03	0.03	0.01	0.00
13	13	42	0.25	0.59	0.15	0.02	0.01	0.01	0.00
14	14	42	0.23	0.65	0.11	0.01	0.01	0.01	0.00
15	15	43	0.00	0.14	0.86	0.00	0.01	0.00	0.01
16	16	109	0.00	0.20	0.80	0.01	0.02	0.00	0.01
17	17	30	0.01	0.17	0.82	0.00	0.01	0.00	0.01
18	18	30	0.02	0.31	0.67	0.01	0.00	0.00	0.00
19	19	143	0.01	0.22	0.77	0.01	0.02	0.00	0.01
20	20	28	0.02	0.40	0.58	0.00	0.00	0.00	0.00
21	21	92	0.09	0.90	0.01	0.03	0.03	0.00	0.00
22	22	11	0.37	0.49	0.14	0.01	0.00	0.01	0.00
23	23	35	0.08	0.91	0.01	0.02	0.01	0.00	0.00
24	1,3,5,7	1,446							
25	2,4,6,8	1,446							
26	19,20	171							
27	15,17	73							
28	16,18	139							
29	9,11,13,22	604							
30	10,12,14,22	630							

APPENDIX C

LISTING OF MAIN COMPUTER PROGRAM

APPENDIX C

LISTING OF MAIN COMPUTER PROGRAM

```
main: procedure options(main);

/*** problem definition statements *************************/

declare
    printintermed           character(80) varying,
    printrandno             character(80) varying,
    didimension             fixed binary,
    partdimension           fixed binary,
    sortiedimension         fixed binary,
    samplesize              fixed binary,
    scalefactor             float binary,
    conffactor              float binary,
    seed                    fixed binary,   /* random number seed */

    outfile                 file,
    outfile_name            char(80) varying,
    1 outrec,
        2 out_i             picture 'zz9',
        2 out_cann1         picture 'zzzz9',
        2 out_cann2         picture 'zzzz9',
        2 out_cann3         picture 'zzzz9',
        2 out_cann4         picture 'zzzz9',
        2 out_cost1         picture 'zzzzzzz9v.9',
        2 out_cost2         picture 'zzzzzzz9v.9',
        2 out_cost3         picture 'zzzzzzz9v.9',
        2 out_cost4         picture 'zzzzzzz9v.9',

    p_accept_part           float binary,

    cost_phigh              float binary,
    cost_low_l              float binary,
    cost_low_u              float binary,
    cost_high_l             float binary,
    cost_high_u             float binary;


get      list(printintermed)      options(prompt('printintermed (Y/N)'));
get skip list(printrandno)        options(prompt('printrandno (Y/N)'));
get skip list(didimension)        options(prompt('didimension'));
get skip list(partdimension)      options(prompt('partdimension'));
get skip list(sortiedimension)    options(prompt('sortiedimension'));
get skip list(samplesize)         options(prompt('samplesize'));
get skip list(scalefactor)        options(prompt('scalefactor'));
get skip list(conffactor)         options(prompt('conffactor'));
get skip list(seed)               options(prompt('seed (-1 for sytem seed)'));
get skip list(outfile_name)       options(prompt('outfile_name'));
get skip list(p_accept_part)      options(prompt('p_accept_part'));
get skip list(cost_phigh)         options(prompt('cost_phigh'));
get skip list(cost_low_l)         options(prompt('cost_low_l'));
get skip list(cost_low_u)         options(prompt('cost_low_u'));
get skip list(cost_high_l)        options(prompt('cost_high_l'));
get skip list(cost_high_u)        options(prompt('cost_high_u'));

put page list('echo problem definition variables');
put skip list(didimension,'didimension');
put skip list(partdimension,'partdimension');
put skip list(sortiedimension,'sortiedimension');
put skip list(samplesize,'samplesize');
put skip list(scalefactor,'scalefactor');
put skip list(conffactor,'conffactor');
put skip list(outfile_name,'outfile_name');
put skip list(p_accept_part,'p_accept_part');
put skip list(cost_phigh,'cost_phigh');
put skip list(cost_low_l,'cost_low_l');
put skip list(cost_low_u,'cost_low_u');
put skip list(cost_high_l,'cost_high_l');
put skip list(cost_high_u,'cost_high_u');

begin;

open file(outfile) title(outfile_name) output record
        environment(fixed_length_records,maximum_record_size(63));


/*** declaration of globals variables *****************************************/

declare
    di(partdimension,didimension)   fixed binary,  /* part-di cross reference */
    c(partdimension)                float binary,  /* cost vector */

    ra(didimension)                 float binary(113),  /* actual rates */
    rp(didimension)                 float binary(113),  /* predicted rates */
    rk(sortiedimension)             float binary(113),  /* damage outcome */

    rnset   external entry (fixed binary),         /* set seed */
    rnget   external entry (fixed binary),         /* get seed */
    rnunf   external entry returns(float binary);  /* random number */


if seed > 0 then call rnset(seed);
call rnget(seed);
put page;
put skip list('random number seed: ',seed);

/******* main routine ********************************************/
/****** monte carlo integration **********************************/

declare
    i                           fixed binary,

    s1(partdimension)           fixed binary,       /* spares level model 1 */
    s2(partdimension)           fixed binary,       /* spares level model 2 */
    s3(partdimension)           fixed binary,       /* spares level model 3 */
    s4(partdimension)           fixed binary,       /* spares level model 4 */
    k(didimension)              fixed binary,       /* damage outcome */
    cann1                       fixed binary,       /* cann ac for model 1 */
    cann2                       fixed binary,       /* cann ac for model 2 */
    cann3                       fixed binary,       /* cann ac for model 3 */
    cann4                       fixed binary,       /* cann ac for model 4 */
    cost1                       float binary,       /* cost of model 1 parts */
    cost2                       float binary,       /* cost of model 2 parts */
    cost3                       float binary,       /* cost of model 3 parts */
    cost4                       float binary;       /* cost of model 4 parts */

if printintermed = 'Y' then put skip(3) list('intermediate results');
```

```
do i = 1 to samplesize;

    call getcostvect;
    call getdimatrix;
    call getdamagevect;
    call s_aso(s1);
    call computecost(s1,cost1);
    call s_rbs(s2,cost1);
    call computecost(s2,cost2);
    call s_asodi(s3);
    call computecost(s3,cost3);
    call s_rbsdi(s4,cost3);
    call computecost(s4,cost4);
    call computek(k);
    call computecann(k,s1,cann1);
    call computecann(k,s2,cann2);
    call computecann(k,s3,cann3);
    call computecann(k,s4,cann4);


    if printintermed = 'Y'
        then
        do;
            put skip(2);
            put skip list('i: ',i);
            put skip list('s1: ',s1);
            put skip list('s2: ',s2);
            put skip list('s3: ',s3);
            put skip list('s4: ',s4);
            put skip list('cost1: ',cost1);
            put skip list('cost2: ',cost2);
            put skip list('cost3: ',cost3);
            put skip list('cost4: ',cost4);
            put skip list('k: ',k);
            put skip list('cann1:',cann1);
            put skip list('cann2:',cann2);
            put skip list('cann3:',cann3);
            put skip list('cann4:',cann4);
        end;

    out_i = i;
    out_cann1 = cann1;
    out_cann2 = cann2;
    out_cann3 = cann3;
    out_cann4 = cann4;
    out_cost1 = cost1;
    out_cost2 = cost2;
    out_cost3 = cost3;
    out_cost4 = cost4;
    write file(outfile) from(outrec);

end;

stop;

/*******************************************/

computek: procedure(k);
```

```
declare
    k(*)                          fixed binary,   /* damage outcome */

    i                             fixed binary,
    j                             fixed binary,
    r                             float binary;

do i = 1 to didimension;
    k(i) = 0;
end;
do i = 1 to sortiedimension;
    r = rk(i);
    j = 0;
    do while(r > 0);
        j = j + 1;
        r = r - ra(j);
    end;
    k(j) = k(j) + 1;
end;

end computek;

/***** compute number of cannibalized  aircraft ******/

computecann: procedure(k,s,nc);

declare
    s(*)                          fixed binary,       /* spares level */
    k(*)                          fixed binary,       /* damage outcome */
    nc                            fixed binary,       /* number of canned ac */

    i                     fixed binary,
    j                     fixed binary,
    dp                    fixed binary;

nc = 0;
do i = 1 to partdimension;
    dp = 0;
    do j = 1 to didimension;
        dp = dp + k(j)*di(i,j);
    end;
    nc = max(nc,dp-s(i));
end;

end computecann;


/****************************/

getdimatrix: procedure;

declare
    acceptdi                              bit(1),
    rvect(partdimension,2:didimension)    float binary,
    i                                     fixed binary,
    j                                     fixed binary;


/* damage-part cross reference di(part,damage incident) */
/* logic assumes damage incident 1 is the 'no damage' event */

do i = 1 to partdimension;
    di(i,1) = 0;
end;

do j = 2 to didimension;
acceptdi = '0'b;
 do while(↑acceptdi);
 do i = 1 to partdimension;
    rvect(i,j) = rnunf();
    if rvect(i,j) < p_accept_part
        then
          do;
            di(i,j)  = 1;
            acceptdi = '1'b;
          end;
        else di(i,j) = 0;
 end;
 end;
end;

if printrandno = 'Y'
    then
            do i = 1 to partdimension;
             j = 1;
             put skip  list('i,j,*,di(i,j)',i,j,di(i,j));
             do j = 2 to didimension;
              put skip  list('i,j,rvect(i,j),di(i,j)',i,j,rvect(i,j),di(i,j));
             end;
            end;

end getdimatrix;

/****************************/

getcostvect: procedure;

/* cost vector c(part) */

declare
    rvect(2,partdimension)  float binary,
    i                       fixed binary;

do i = 1 to partdimension;
  rvect(1,i) = rnunf();
  rvect(2,i) = rnunf();
end;

do i = 1 to partdimension;
    if rvect(1,i) < cost_phigh
    then
        c(i) = cost_high_l + (cost_high_u-cost_high_l)*rvect(2,i);
    else
        c(i) = cost_low_l + (cost_low_u-cost_low_l)*rvect(2,i);
end;

if printrandno = 'Y'
    then
```

```
        do i = 1 to partdimension;
            put skip list('i,r(*,i),cost(i): ',i,rvect(1,i),rvect(2,i),c(i));
        end;

end getcostvect;

/****** random damage rate vector generator ***********/

getdamagevect: procedure;

/* logic assumes damage incident 1 corresponds to no damage */

declare
    rvect(didimension-1)     float binary,
    i                        fixed binary;

/* rp */

do i = 1 to didimension-1;
    rvect(i) = rnunf();
end;
rp(1) = 1;
do i = 2 to didimension;
    rp(i) = scalefactor*rvect(i-1);
    rp(1) = rp(1) - rp(i);
end;

if printrandno = 'Y'
    then
        do;
            put skip list('rp(1)',rp(1));
            do i = 2 to didimension;
                put skip list('i,r,rp: ',i,rvect(i-1),rp(i));
            end;
        end;


/* ra = (1-conffactor)*ra + conffactor*rp) */

do i = 1 to didimension-1;
    rvect(i) = rnunf();
end;
ra(1) = 1;
do i = 2 to didimension;
    ra(i) = scalefactor*rvect(i-1);
    ra(i) = (1-conffactor)*ra(i) + conffactor*rp(i);
    ra(1) = ra(1) - ra(i);
end;

if printrandno = 'Y'
    then
        do;
            put skip list('ra(1)',ra(1));
            do i = 2 to didimension;
                put skip list('i,r,ra: ',i,rvect(i-1),ra(i));
            end;
        end;
```

```
/* rk */

do i = 1 to sortiedimension;
    rk(i) = rnunf();
end;

if printrandno = 'Y'
    then
        do i = 1 to sortiedimension;
            put skip list('i,rk: ',i,rk(i));
        end;

end getdamagevect;

/***********************************************************/
/* aso spares policy implementation */

s_aso: procedure(s);

%replace safetyfactor by 0.9;

declare
    s(*)                            fixed binary,     /* spares level */

    r_part(partdimension)           float binary(113), /* part damage rates */

    pn  fixed binary,
    j   fixed binary,
    mm  float binary(113),
    bb  float binary(113),
    tt  float binary(113);

/* build up damage rates for parts */

do pn = 1 to partdimension;
    r_part(pn) = 0;
    do j = 1 to didimension;
        r_part(pn) = r_part(pn) + di(pn,j)*rp(j);
    end;
end;

/* built aso stock level */

do pn = 1 to partdimension;
    if r_part(pn) ¬= 1
    then
        do;
            mm = r_part(pn)/(1-r_part(pn));
            bb = (1-r_part(pn))**sortiedimension;
            tt = bb;
            s(pn) = 0;
            do while(bb < safetyfactor);
                s(pn) = s(pn) + 1;
                tt = tt*mm*(sortiedimension+1-s(pn))/s(pn);
                bb = bb + tt;
            end;
        end;
    else
        do;
```

```
            put skip list('trivial case...',r_part);
            stop;
        end;
end;

end s_aso;

/***************************************************/


/***********************************************************/
/* aso damage incident spares policy implementation */

s_asodi: procedure(s);

%replace safetyfactor by 0.9;

declare
    s(*)                                 fixed binary,      /* spares level */

    dis(2:didimension)  fixed binary,
    din                 fixed binary,
    j                   fixed binary,
    mm                  float binary(113),
    bb                  float binary(113),
    tt                  float binary(113);


/* built aso damage incident stock level */
/* assumes damage incident 1 is the 'no damage' event */

do din = 2 to didimension;
    if rp(din) ↑= 1
    then
        do;
            mm = rp(din)/(1-rp(din));
            bb = (1-rp(din))**sortiedimension;
            tt = bb;
            dis(din) = 0;
            do while(bb < safetyfactor);
                dis(din) = dis(din) + 1;
                tt = tt*mm*(sortiedimension+1-dis(din))/dis(din);
                bb = bb + tt;
            end;
        end;
    else
        do;
            put skip list('trivial case...',rp);
            stop;
        end;
end;

do j = 1 to partdimension;
    s(j) = 0;
    do din = 2 to didimension;
        s(j)  = s(j) + di(j,din)*dis(din);
    end;
end;
```

```
end s_asodi;

/**********************************************/

/**********************************************/
/* rbs spares policy implementation */

s_rbs: procedure(s,costgoal);

declare
    s(*)                            fixed binary,
    costgoal                        float binary,

    r_part(partdimension)           float binary(113),

    j                       fixed binary,
    mm(partdimension)       float binary(113),
    bb(partdimension)       float binary(113),
    tt(partdimension)       float binary(113),
    pn                      fixed binary,

    bestpn                  fixed binary,
    bestratio               float binary(113),
    currdelta               float binary(113),
    currratio               float binary(113),
    cost                    float binary;

/* initialize */

cost = 0;

/* build up damage rates for parts */

do pn = 1 to partdimension;
    r_part(pn) = 0;
    do j = 1 to didimension;
        r_part(pn) = r_part(pn) + di(pn,j)*rp(j);
    end;
end;

do pn = 1 to partdimension;
    s(pn) = 0;
    if r_part(pn) ↑= 1
        then
            do;
                mm(pn) = r_part(pn)/(1-r_part(pn));
                bb(pn) = (1-r_part(pn))**sortiedimension;
                tt(pn) = bb(pn);
            end;
        else
            do;
                put skip list('Trivial problem: ',r_part);
                stop;
            end;
end;

/* build rbs level */
```

```
do while(cost < costgoal);
    currdelta = tt(1)*mm(1)*(sortiedimension-s(1))/(s(1)+1);
    bestpn = 1;
    bestratio = currdelta/(c(1)*bb(1));
    do pn = 2 to partdimension;
            currdelta = tt(pn)*mm(pn)*(sortiedimension-s(pn))/(s(pn)+1);
            currratio = currdelta/(c(pn)*bb(pn));
            if currratio > bestratio
                then
                    do;
                        bestpn = pn;
                        bestratio = currratio;
                    end;
    end;
    cost = cost + c(bestpn);
    if cost <= costgoal
        then
        do;
            tt(bestpn) = tt(bestpn)*mm(bestpn)*(sortiedimension-s(bestpn))/(s(bestpn)+1);
            bb(bestpn) = bb(bestpn) + tt(bestpn);
            s(bestpn) = s(bestpn) + 1;
        end;
end;

end s_rbs;

/*********************************************************************/

/***********************************************/
/* rbs damage incident policy implementation */

s_rbsdi: procedure(s,costgoal);

declare
    s(*)                            fixed binary,
    costgoal                        float binary,

    dis(2:didimension)              fixed binary,

    j                       fixed binary,
    mm(2:didimension)       float binary(113),
    bb(2:didimension)       float binary(113),
    tt(2:didimension)       float binary(113),
    din                     fixed binary,

    bestdin                 fixed binary,
    bestratio               float binary(113),
    currdelta               float binary(113),
    currratio               float binary(113),
    cost                    float binary,
    cdi(2:didimension)      float binary;

/* initialize */

cost = 0;

/* initialize cdi  */

do din = 2 to didimension;
```

```
    cdi(din) = 0;
    do j = 1 to partdimension;
        cdi(din) = cdi(din) + di(j,din)*c(j);
    end;
end;

/******/
/* assumes damage incident 1 is the 'no damage' event */

do din = 2 to didimension;
    dis(din) = 0;
    if rp(din) ↑= 1
        then
            do;
                mm(din) = rp(din)/(1-rp(din));
                bb(din) = (1-rp(din))**sortiedimension;
                tt(din) = bb(din);
            end;
        else
            do;
                put skip list('Trivial problem: ',rp);
                stop;
            end;
end;

/* build rbsdi level */

do while(cost < costgoal);
    currdelta = tt(2)*mm(2)*(sortiedimension-dis(2))/(dis(2)+1);
    bestdin = 2;
    bestratio = currdelta/(cdi(2)*bb(2));
    do din = 2 to didimension;
            currdelta = tt(din)*mm(din)*(sortiedimension-dis(din))/(dis(din)+1);
            currratio = currdelta/(cdi(din)*bb(din));
            if currratio > bestratio
                then
                    do;
                        bestdin = din;
                        bestratio = currratio;
                    end;
    end;
    cost = cost + cdi(bestdin);
    if cost <= costgoal
        then
        do;
            tt(bestdin) = tt(bestdin)*mm(bestdin)*(sortiedimension-dis(bestdin))/(dis(bestdin)+1);
            bb(bestdin) = bb(bestdin) + tt(bestdin);
            dis(bestdin) = dis(bestdin) + 1;
        end;
end;

do j = 1 to partdimension;
    s(j) = 0;
    do din = 2 to didimension;
        s(j)  = s(j) + di(j,din)*dis(din);
    end;
end;

end s_rbsdi;
```

```
/**********************************************************************/

computecost: procedure(s,cost);

declare
    i                            fixed binary,
    s(*)                         fixed binary,
    cost                         float binary;

/* compute cost  */
cost = 0;
do i = 1 to partdimension;
    cost = cost + s(i)*c(i);
end;

end computecost;

/************************************************************************/

end; /* begin */

end main;
```